THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 7, September 1982
Printed in U.S.A.

Piecewise Linear Approximation of Multivariate Functions

By G. S. LEE

(Manuscript received January 5, 1982)

To approximate functions of a single variable by using linear interpolation is routine in empirical studies. Here, we consider approximating functions of several *variables in a similar piecewise linear manner. We focus on the nontrivial part of this technique, which is that of choosing the appropriate "pieces" for the piecewise linear approximation. Precisely, we seek to identify the best interpolants to use at a point of interpolation. This is not an issue for functions of a single variable, since the linear ordering of the number line leaves us with no choice. For functions of several variables, we propose several simple tools to help uncover undesirable choices. The techniques presented are useful in the empirical study of quantitatively complex functional relationships whose qualitative behavior is nevertheless known and simple. Response time relationships parametrized by workloads in computer performance modeling often fall into this category, and an actual bivariate function of this type is used to motivate the development.*

I. INTRODUCTION

Linear approximation is a popular and economical way to gain appreciation of the behavior of functional relationships. Especially in higher dimensions, making sense out of a sample of data points in terms of the underlying multivariate relationship is greatly facilitated by some form of piecewise linear approximation. Such an approximation allows for estimation of the function at values not included in the sample, sheds light on the activity of the function at selected neighborhoods, and identifies regions where the relationship behaves interestingly.

For the sake of concreteness let us consider a typical problem. A simulation model of a computer system has been given. The workload driving the system is described by two variables giving the respective

percentages of two classes of users in the user population; there are three classes of users altogether. The two variables, then, form a bivariate parametrization of the workload. Six simulations were run, and the resulting mean response times are shown below. (This response time function will be denoted by h throughout this paper.)

$$h(60, 25) = 5.5$$

$$h(60, 7) = 2.1$$

$$h(40, 30) = 1.2$$

$$h(40, 7) = 1.0$$

$$h(20, 25) = 0.7$$

$$h(20, 15) = 0.7$$

It will be necessary to estimate the response times for many more workloads than for the six simulations already run. Yet, it would be inefficient to make a run for each possible parameter pair (x_1, x_2). It would also not be necessary, given the qualitatively simple relationship that generally exists between workloads and response times of computer systems. In this particular case, the function is expected to be monotone. Therefore, the six simulations not only give us the values of the function at those six points, but also the values between and around them, that is, at least approximately. For example, we could safely assert that $h(55, 10)$ should lie between 1.0 and 5.5. In fact, it would be reasonable to estimate the range to be from 1.8 to 5.5. This becomes obvious by plotting the six-parameter pairs used in the simulations, plus the point $P = (55, 10)$ on the (x_1, x_2) plane, as shown in Fig. 1. The points A through F are labeled in tbe order in which they were collected; point A is the oldest. Since the polygon with corners A, D, B, and E bound the point (55, 10), it is reasonable to deduce that the values of h at A, D, B, and E bound the value of h at (55, 10). This gives the interval (1.0, 5.5). Furthermore, $h(55, 7)$ may be estimated as 1.8 by linear interpolation between $h(40, 7) = 1.0$ and $h(60, 7) = 2.1$. Since h is increasing in both x_1 and x_2, $h(55, 10) \geqq h(55, 7)$. This gives the sharper interval (1.8, 5.5).

As we will see, this approacb can be continued until a numerical estimate for $h(55, 10)$ is reached. Two steps are required. The first step identifies the data points that contain relevant information about $h(55, 10)$. For example, we have already rejected C and F as irrelevant to P. This was simple and was done "by eye." Finer methods need be developed, however, to determine which one of A, B, D or E should be further discarded. The major part of this paper, starting witb Section III, deals with this and related problems. The second step consists of fitting a linear function over the data points chosen as a result of the

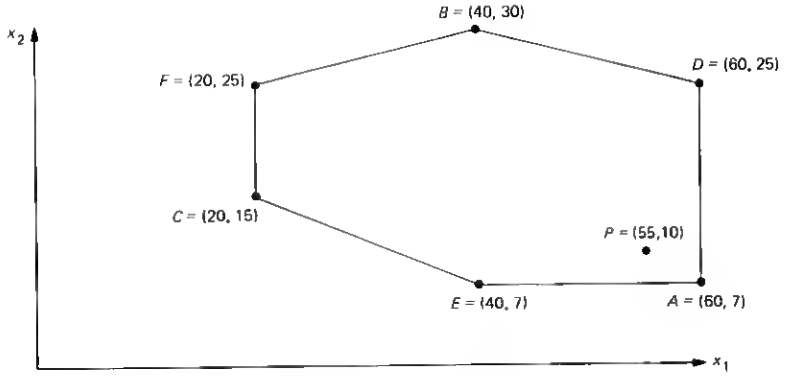

Fig. 1—Data points for case study.

first step; the mechanics of linear fitting will be reviewed in Section II. Together, the two steps produce the "best" way to fit the data points in a piecewise linear way. The concluding recapitulation gives an overview of the methodology in an iterative context.

Formulas used in our treatment are gathered in a sequence of propositions. Since they are rather straightforward and our interest is in their usage, only abbreviated arguments for their validity are included. All calculations may be easily carried out on a computer. Needed, aside from basic arithmetic, are routines to define and manipulate matrices, perform matrix multiplications and inversions, take determinants, and find eigenvectors of symmetric matrices. Computations for our examples have been carried out with relatively short programs in the *S* statistical package.[1]

Our proposed approach is not meant to replace the standard practice of fitting a single, global functional relationship to the data. The two procedures reveal different aspects of the data. Function fitting, being inherently global, is well suited for capturing the overall behavior of the relationship. This we hope to complement with the piecewise linear approach, which, being inherently local, is better suited for pinpointing places where interesting things happen to the function. This is especially relevant for iterative empirical studies, where it is useful to know where the function is active in the parameter space so that further experiments can be fruitfully specified.

II. THE BASIC INTERPOLATION

Linear interpolation for functions of a single variable is usually taught in high school in terms of such notions as similar triangles and slopes. For the purpose of generalization to higher dimensions, an alternative, though algebraically equivalent, viewpoint is preferred.

Specifically, let $f(x_1) = y_1$, $f(x_2) = y_2$, and $x_1 < x < x_2$. It can be shown that the usual linear interpolation gives the estimate

$$\alpha_1 y_1 + \alpha_2 y_2,$$

where α_1 and α_2 satisfy

$$\alpha_1 x_1 + \alpha_2 x_2 = x,$$

$$\alpha_1 + \alpha_2 = 1,$$

$$\alpha_1, \alpha_2 > 0.$$

In other words, to estimate $f(x)$ for some x in the interval $x_1 < x_2$, express x as a convex combination (weighted average) of x_1 and x_2, then estimate $f(x)$ as the same convex combination of $f(x_1)$ and $f(x_2)$.

Notations: As usual, $\mathbf{R}^n$ is the set of n-dimensional real column vectors. Also standard is the use of the prime notation for matrix transposition, as in $\mathbf{M}'$. If the jth column of matrix $\mathbf{M}$ is $\mathbf{m}_j$, we write $\mathbf{M} = (\mathbf{m}_1, \cdots, \mathbf{m}_k)$. The length of a vector $\mathbf{v} = (v_1, \cdots, v_n)'$ is denoted by

$$\|\mathbf{v}\| = \sqrt{(v_1^2 + \cdots + v_n^2)}.$$

For any $\mathbf{x} \in \mathbf{R}^n$, let $\mathbf{x}^* \in \mathbf{R}^{n+1}$ denote

$$\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix},$$

i.e., the vector consisting of $\mathbf{x}$ followed by the singleton 1. If f is a function, $\hat{f}$ denotes its approximation by linear interpolation.

A higher dimensional analogue of having the points x_1, x_2 generate an interval in one dimension is having points $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ generate a triangle in two dimensions and having $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$ generate a pyramid in three dimensions. Just as we require that an interval in one dimension should not shrink to a point, we require that a triangle should contain area on a plane rather than reduce to a line segment, and that a pyramid should contain volume in solid space rather than collapse onto some plane. In general, we are concerned with the convex hull of $(n + 1)$ points $\mathbf{x}_1, \cdots, \mathbf{x}_{n+1} \in \mathbf{R}^n$, subject to the condition that these points do not lie in some lower ($<n$) dimensional hyperplane. The matrix formulation of this geometric requirement is that the linear transformation $\mathbf{X}$ given by

$$\mathbf{X} = (\mathbf{x}_1 - \mathbf{x}_{n+1}, \cdots, \mathbf{x}_n - \mathbf{x}_{n+1})$$

should have the trivial kernel, or, by the rank-nullity theorem, satisfy $\det(\mathbf{X}) \neq 0$.

Definition: An n-dimensional pyramid is the convex hull of a set of $(n + 1)$ vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$ such that

$$\det(\mathbf{x}_1 - \mathbf{x}_{n+1}, \cdots, \mathbf{x}_n - \mathbf{x}_{n+1}) \neq 0.$$

The points $\mathbf{x}_i$ are called its vertices, and any subset of n vertices forms a face of the pyramid.

We will often call a set of points a pyramid, rather than a convex hull generated by these points, as would be strictly correct; however, this makes no difference since there is a one-to-one correspondence between the set of vertices and the pyramid it generates.

Proposition 1: Let f: $\mathbf{R}^n \to \mathbf{R}$ be a function of n variables. Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$ be (i.e., generate) an n-dimensional pyramid containing $\mathbf{x} \in \mathbf{R}^n$, and let $f(\mathbf{x}_j) = y_j$. Then,

(*i*) $\mathbf{X}^* = (\mathbf{x}_1^*, \cdots, \mathbf{x}_{n+1}^*)$ is an invertible $(n+1) \times (n+1)$ matrix, and

(*ii*) the estimate $\hat{f}(\mathbf{x}) = (y_1, \cdots, y_{n+1})(\mathbf{X}^*)^{-1}\mathbf{x}^*$ is a higher dimensional generalization of linear interpolation for a single variable.

Proof:

(*i*) If $\mathbf{X}^*$ were not invertible, its columns would be linearly dependent. Therefore, one of the vertices would be a convex (not necessarily positive) combination of the n remaining vertices, implying that the $n+1$ points lie in $(n-1)$-dimensional hyperplane at most. This contradicts the definition of a pyramid.

(*ii*) Let $\alpha = (\mathbf{X}^*)^{-1}\mathbf{x}^* = (\alpha_1, \cdots, \alpha_{n+1})'$. Since $\mathbf{x}^* = \mathbf{X}^*\alpha$, it follows from the star (*) definition that

$$\mathbf{x} = \alpha_1\mathbf{x}_1 + \cdots + \alpha_{n+1}\mathbf{x}_{n+1},$$

$$\alpha_1 + \cdots + \alpha_{n+1} = 1.$$

Thus, α expresses $\mathbf{x}$ as a convex combination of $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\}$. Since $\mathbf{x}$ is contained in the convex hull, $\alpha \geqq \mathbf{0}$. Hence, the linear interpolation at $\mathbf{x}$ should be

$$\hat{f}(\mathbf{x}) = \alpha_1 y_1 + \cdots + \alpha_{n+1} y_{n+1},$$

which is as proposed.

In practice, we will not know whether $\mathbf{x}$ is contained in the convex hull. Therefore, the vector of coefficients α should be explicitly computed to check that $\alpha \geqq \mathbf{0}$. The use of the procedure with negative components in α corresponds to linear extrapolation.

Example: Returning to the computer system response time function h, we see from tbe (x_1, x_2) plot in tbe previous section tbat

$$\{B, E, A\} = \left\{\begin{pmatrix}40\\30\end{pmatrix}, \begin{pmatrix}40\\7\end{pmatrix}, \begin{pmatrix}60\\7\end{pmatrix}\right\}$$

forms a two-dimensional pyramid, or triangle, containing

$$P = \begin{pmatrix}55\\10\end{pmatrix}.$$

Alternatively, if plots are not feasible, as would be the case in higher dimensions, we calculate

$$\det\begin{pmatrix} 40-60 & 40-60 \\ 30-\ 7 & 7-\ 7 \end{pmatrix} = 460 \neq 0,$$

checking that a noncollapsing pyramid is obtained, and

$$\alpha = \begin{pmatrix} 40 & 40 & 60 \\ 30 & 7 & 7 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 55 \\ 10 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.13 \\ 0.12 \\ 0.75 \end{pmatrix} \geqq \mathbf{0},$$

checking that

$$\begin{pmatrix} 55 \\ 10 \end{pmatrix}$$

is contained in the pyramid. Interpolation according to the formula gives

$$(1.2 \quad 1.0 \quad 2.1) \begin{pmatrix} 0.13 \\ 0.12 \\ 0.75 \end{pmatrix} = 1.85.$$

Similarly,

$$\{E, A, D\} = \left\{ \begin{pmatrix} 40 \\ 7 \end{pmatrix}, \quad \begin{pmatrix} 60 \\ 7 \end{pmatrix} \quad \begin{pmatrix} 60 \\ 25 \end{pmatrix} \right\}$$

is a pyramid containing

$$\begin{pmatrix} 55 \\ 10 \end{pmatrix},$$

and interpolation using this pyramid gives

$$(1.0 \quad 2.1 \quad 5.5) \begin{pmatrix} 40 & 60 & 60 \\ 7 & 7 & 25 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 55 \\ 10 \\ 1 \end{pmatrix} = 2.39.$$

III. THE SELECTION PROBLEM

We have just seen how two different choices of pyramids can lead to two markedly different approximations. Taking the average and approximating

$$\hat{h} \begin{pmatrix} 55 \\ 10 \end{pmatrix}$$

by $\frac{1}{2}(1.85 + 2.39) = 2.12$ is not justified. An average is appropriate when summarizing a batch of numbers that have been made different by random error, since the averaging process "zeroes out" the random errors. In our case, the difference between the two approximations

does not arise from random error. One of the choices is better than the other, and that is not a probabilistic phenomenon.

Another way to proceed is to put greater trust on closer interpolants than on further ones and choose the pyramid yielding the least-summed distance from its vertices to that point. Under this criterion, the pyramid ADE should be preferred over ABE, since D is closer to P than B is by nine units. Unfortunately, the notion of distance is not invariant to the choice of scales on the axes, and the criterion becomes sensitive to changes of units in the independent variables. The distance-type criteria have a further, more fundamental weakness, which they share with, among others, the criterion of Lawson.[2]

As used by Akima[3] in his algorithm for bivariate smooth interpolation, Lawson's criterion triangulates the (x_1, x_2) plane in such a way that minimum interior angles of triangles are maximized. The objective is to set up as many "fat" triangles as possible. Under such "fatness-type" criterion, ADE is again chosen over ABE, since ABE comes with the companion triangle ABD, which is too "skinny." An extremely skinny triangle is undesirable for two reasons. First, its $\mathbf{X}^*$ may be numerically unstable to invert. Second, it biases the estimation along a particular direction in the (x_1, x_2)-plane. (On the other hand, we will see in Section VI that thoughtful biasing may be beneficial.) It is not necessary, however, to make fatness an overall criterion just to avoid such excesses.

The problem with these distance- and fatness-type criteria is that only the configuration of the $\mathbf{x}$ variables is used in assessing pyramids. (With the curve-fitting approach that uses Lagrangian and trigonometric polynomials,[4] even this information is not taken into account.) This is an unnecessary restriction. Surely the values of the function at the interpolants, the $f(\mathbf{x})$'s, have much to say on the adequacy of an interpolation. Our basic working tool to tie $\mathbf{x}$ together with $f(\mathbf{x})$ is the notion of steepest ascent. Using it, we show that the correct pyramid to choose is ABE, rather than the pyramid ADE, favored by both criteria above. Incidentally, the ABE choice for (55, 10) was confirmed empirically by an actual simulation run which gave the result $h(55, 10) = 1.74$. Recall that interpolation with ABE gives 1.85, representing a 6.3 percent error, while interpolation with ADE gives 2.39, representing a 37.4 percent error. The fact that the interpolation overestimates could also have been predicted. See Section VII, where assumptions underlying our approach are discussed.

IV. STEEPEST ASCENT

Equipped with the mechanics of basic interpolation, we may analyze in detail the function's interpolated behavior on a given pyramid. We would like to know along which direction inside the pyramid the

function increases, and how fast that increase is. Because of the linearity of the interpolation, both questions have well-defined answers. The geometric way to derive an answer is to slice the pyramid into parallel "contour hyperplanes," which are systems of hyperplanes such that points on the same hyperplane have the same interpolated values.

Example: Figure 2 shows some contour hyperplanes (contour lines in two dimensions) corresponding to the pyramid (triangle) *ABE*. Points along the downward-sloping lines take the same interpolated values as labeled. Clearly, $\hat{h}$ increases most rapidly along the direction of the arrow, which is perpendicular to the contour lines.

Since the contour hyperplanes are parallel, a single vector perpendicular to all of them can be found. Actually, there will be many such vectors, but they can only have one of two opposite directions, one for increasing f and another for decreasing f. We agree to take the increasing direction. Being perpendicular to the contour hyperplanes, this direction has no component along which $\hat{f}$ does not increase. Therefore, it is the direction along which $\hat{f}$ increases the fastest. We define a unit vector in this direction to be the direction of steepest ascent of the function f in the pyramid. The rate of steepest ascent is the increase of $\hat{f}$ along the direction of steepest ascent per unit distance traveled.

It remains to formulate these geometric notions in matrix terms.

Proposition 2: Let f: $\mathbf{R}^n \to \mathbf{R}$ be a function of n variables, $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$ be a pyramid, $f(\mathbf{x}_i) = y_i$, and $\mathbf{X}^* = (\mathbf{x}_1^*, \cdots, \mathbf{x}_{n+1}^*)$. Let

$$(a_1, \cdots, a_{n+1}) = (y_1, \cdots, y_{n+1})(\mathbf{X}^*)^{-1}.$$

(*i*) The direction of steepest ascent in $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\}$ is $(a_1, \cdots, a_n)/\|(a_1, \cdots, a_n)\|$, and

(*ii*). The rate of steepest ascent in $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\}$ is $\|(a_1, \cdots, a_n)\|$.

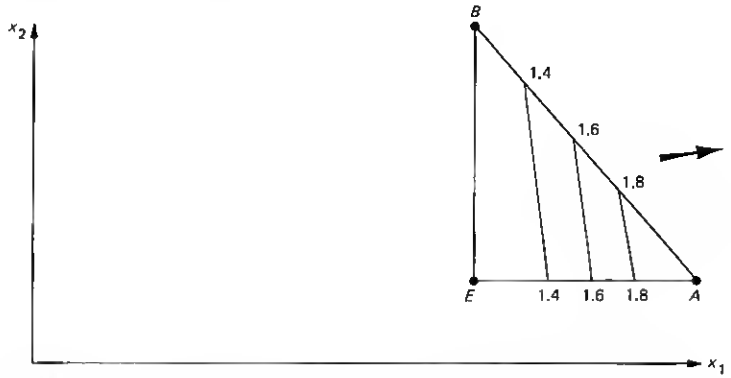

Fig. 2—Contour lines for *ABE*. Function increases in direction of arrow.

Proof: By definition of $(a_1, \cdots, a_{n+1})$, the interpolation fits the function

$$\hat{f}(x_1, \cdots, x_n) = a_1 x_1 + \cdots + a_n x_n + a_{n+1}$$

on the pyramid $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\}$.
To show (i), note that $\hat{f}(\mathbf{u}) = \hat{f}(\mathbf{v})$ implies that $(a_1, \cdots, a_n)(\mathbf{u} - \mathbf{v}) = 0$. To show (ii), note that $\hat{f}((a_1, \cdots, a_n)/\|(a_1, \cdots, a_n)\|) - \hat{f}(\mathbf{0}) = \|(a_1, \cdots, a_n)\|$.
Example: Table I lists some directions and rates of steepest ascent that we will use later.

V. A FIRST APPROACH: JUDGING BY FACES

Example: We are now ready to reject *ADE* in favor of *ABE*. Actually, since *ADE* and *BDE* form a companion pair, we will reject them both. This will be accomplished by analyzing the face *DE*. In Fig. 3, *ADE* and *BDE* are shown along with their respective directions of steepest ascent loosely placed about their centers. The two directions of steepest ascent make sense separately. Both show response time as an increasing function of x_1 and x_2. Together, however, they show that *DE* is a ridge, i.e., $\hat{h}$ increases as *DE* is approached from either side.

Table I—Some directions and rates of steepest ascent

Pyramid	Direction of Steepest Ascent	Rate of Steepest Ascent
ADE	(0.29, 0.96)′	0.197
BDE	(0.99, 0.05)′	0.218
ABD	(0.81, 0.58)′	0.323
ABE	(0.98, 0.19)′	0.059

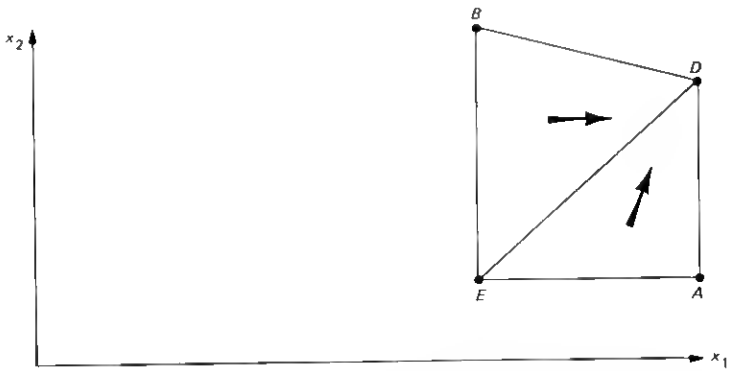

Fig. 3—ADE and BDE with their directions of steepest ascent.

Alternatively, a ridge indicates that $\hat{h}$ is not monotone along some direction on the (x_1, x_2) plane. From our knowledge of h, no such ridge should exist. The triangulation $\{ADE, BDE\}$ incorrectly shows a ridge because $h(D)$ is inordinately large. The triangulation allows D to pull up both triangles, causing their boundary to buckle. The correct way to picture h is shown in Fig. 4. For low values of x_1 and x_2, $\hat{h}$ grows moderately (rate of steepest ascent = 0.059 on ABE), with x_1 being mainly responsible for the increase. As x_1 and x_2 become large, $\hat{h}$ explodes (rate of steepest ascent = 0.323 on ABD), and x_2 begins to affect $\hat{h}$ seriously, although x_1 is still the major contributor. Because of the piecewise linearity of interpolation, AB has become an accelerating, refracting boundary but not a ridge. In Section VI, we will see another reason why ADE is a poor choice.

Note that specific knowledge about h was invoked to make the selection: We knew that response time as a function of workload had no ridges. In general, linear interpolation of functions with known major ridges, valleys, maxima, or minima should be avoided. An exception may be made if we are able to sprinkle such tricky terrain generously with further interpolants. Under finer resolution, the volatile formations should diminish. Of course, even if a ridge does exist, it is highly improbable that it should coincide with one of our faces.

Definition: A face is a bad face if there are two pyramids containing it, and their directions of steepest ascent either both point towards it or both point away from it.

Thus, we accept as likely those faces that may refract and/or accelerate steepest ascent vectors, but we question faces that gather or scatter them.

Matrix techniques are clearly needed to identify bad faces in higher dimensions, where graphs are not feasible. It is enough to find a technique to determine if a given steepest ascent approaches a given

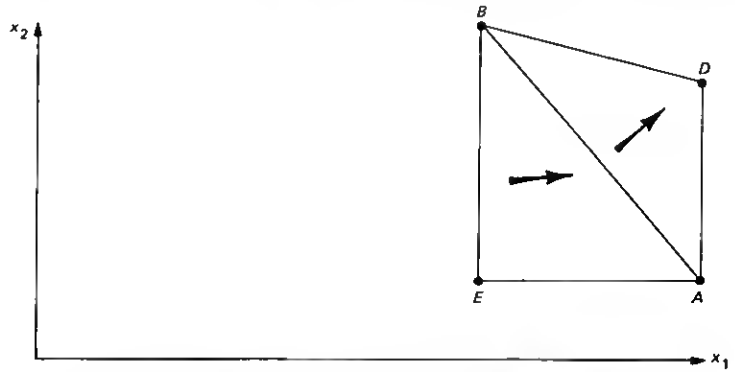

Fig. 4—ABE and ABD with their directions of steepest ascent.

face. Since approaching a face at 90 degrees is different from approaching it at 0.0001 degree, the angle of approach is also of interest. Note that knowledge of the angle alone does not tell us whether the ascent approaches; the ascent may be located on either side of the face.

Proposition 3: Let $\mathbf{a} \in \mathbf{R}^n$ be a steepest ascent for the pyramid $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$. Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\} = F$ be the face of interest. Define the $n \times (n-1)$ matrix

$$\mathbf{X} = (\mathbf{x}_1 - \mathbf{x}_n, \mathbf{x}_2 - \mathbf{x}_n, \cdots, \mathbf{x}_{n-1} - \mathbf{x}_n).$$

Also define the $n \times n$ matrix

$$\mathbf{Y} = (-\mathbf{X}, \mathbf{a}) = (\mathbf{x}_n - \mathbf{x}_1, \mathbf{x}_n - \mathbf{x}_2, \cdots, \mathbf{x}_n - \mathbf{x}_{n-1}, \mathbf{a}).$$

(*i*) Let $\beta = (\beta_1, \cdots, \beta_n)' = Y^{-1}(\mathbf{x}_n - \mathbf{x}_{n+1})$. If $\beta_n > 0$, $\mathbf{a}$ approaches F. If $\beta_n < 0$, $\mathbf{a}$ recedes from F. If $\mathbf{Y}$ is not invertible, $\mathbf{a}$ is parallel to F.

(*ii*) Let $\cos\theta = \sqrt{\mathbf{a}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{a}}$. Then θ is the angle between $\mathbf{a}$ and F.

Proof:

(*i*) The trick here is to pull the vertex $\mathbf{x}_{n+1}$ to the origin so that we may examine the relationship between $\mathbf{a}$, now sitting at the origin, and F, now translated to $\{\mathbf{x}_1 - \mathbf{x}_{n+1}, \cdots, \mathbf{x}_n - \mathbf{x}_{n+1}\}$. For β as given above, let $\pi_i = \beta_i$ for $i = 1, \cdots, n-1$, $\pi_n = 1 - (\pi_1 + \cdots + \pi_{n-1})$, and $t = \beta_n$. The reader may check that

$$t\mathbf{a} = \sum_{i=1}^{n} \pi_i(\mathbf{x}_i - \mathbf{x}_{n+1}),$$

$$\sum_{i=1}^{n} \pi_i = 1.$$

The interpretation of the sign of β_n follows from the trick just described.

(*ii*) The trick is to translate the face to the origin by $\mathbf{x}_n$. We then regress $\mathbf{a}$ against the translated face $\{\mathbf{x}_1 - \mathbf{x}_n, \cdots, \mathbf{x}_{n-1} - \mathbf{x}_n\}$ so that its projection proj($\mathbf{a}$) onto the linear subspace spanned by the translated face may be found by the usual formula. Now take the inner product between $\mathbf{a}$ and proj($\mathbf{a}$), bearing in mind that $\mathbf{a}$ is a unit vector.

Example: Using the formulas above, we find the DE is approached by the direction of steepest ascent of BDE at 31 degrees, and DE is also approached by the direction of steepest ascent of ADE at 39 degrees. Thus, DE is a ridge.

VI. ANOTHER APPROACH: THE AXES CRITERION

Example: Let's consider again the selection problem with the response time function $y = h(x_1, x_2)$, but this time we consider the complementary pair of triangles ABD and BDE, shown in Fig. 5. While neither triangle is very skinny, we could roughly identify directions along

which they stretch out. For example, *ADB* stretches out in some direction along *AB*, which is quite perpendicular to its direction of steepest ascent. *BDE*, on the other hand, stretches somewhat along *DE*, which is not at all perpendicular to its direction of steepest ascent. This difference affords another basis for selection. Let's exaggerate somewhat and give the two triangles more definitive and mutually perpendicular directions of stretch; we also assume that the "direction of steepest ascent of the function" (a vague notion), denoted by **a**, is perpendicular to the stretch of *ABD*. The situation is shown in Fig. 6. We may now rotate the configuration to new axes (x_1', x_2') so that the pair of triangles stretch in directions parallel to the new axes, as in Fig. 7. Because of our assumption, **a** becomes parallel to x_1'. This means that x_2' has no effect on y, so we may plot y as a function of x_1' alone.

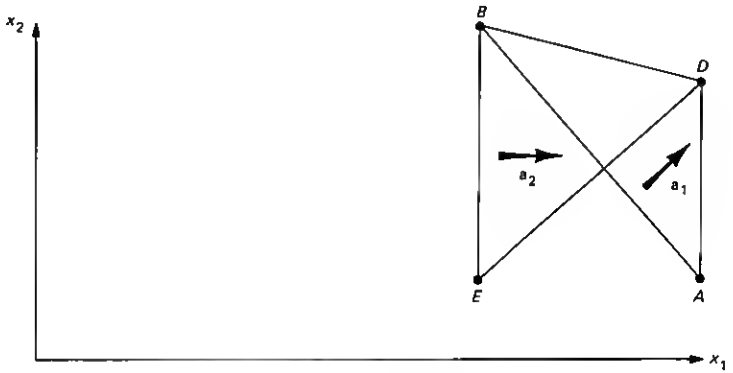

Fig. 5—Triangles *ABD* and *BDE* with directions of steepest ascent.

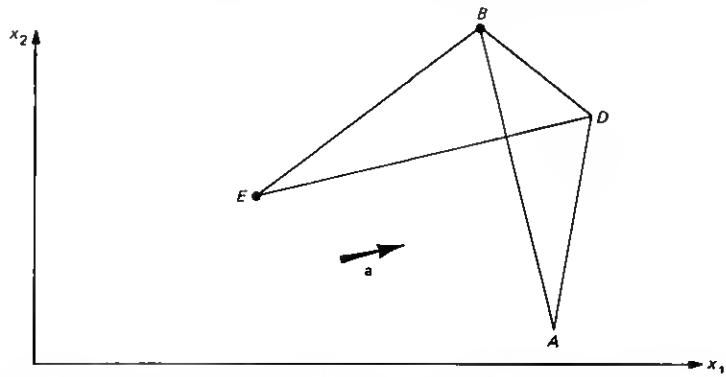

Fig. 6—ABD and BDE stretched in mutually perpendicular directions.

Since a points along x_1', y is an increasing function of x_1'. If y is a linear function of x_1', it will also be a linear function of (x_1, x_2), in which case it doesn't matter which triangle we select. Both will give perfect fit. However, for h, we must protect ourselves against nonlinearity. We actually know that h is concave downwards as a function of x_1'. (If h is concave upwards, the same argument below holds.) The elimination of x_2' leads to the (x_1', y) plot given in Fig. 8. Clearly, ABD hugs the function much better than BDE. By stretching linearly far along the direction of ascent, BDE loses touch of the underlying, nonlinear function. Here is the second reason why ABE is a better choice than ADE.

The moral is that triangles perpendicular to their directions of steepest ascent give better estimates. The same is true in higher dimensions, although in higher dimensions it is not appropriate to talk about the direction of stretch of a pyramid because a pyramid may stretch in several directions at once. In three dimensions, for example,

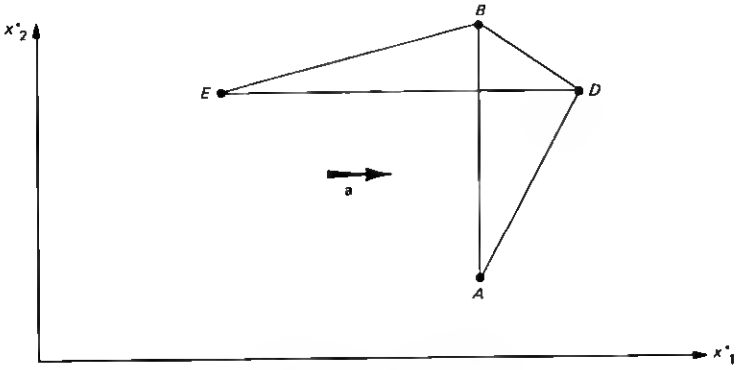

Fig. 7—Rotation of Fig. 6.

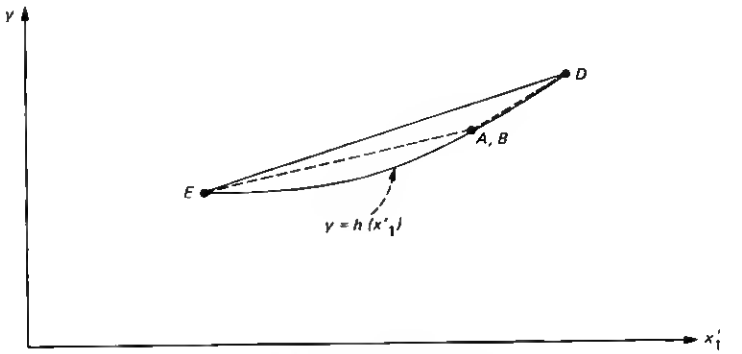

Fig. 8—Elimination of x_2.

a solid pyramid may stretch out in one single direction, as in a needle-like pyramid, or in two directions, as in a pancake-like pyramid. Therefore, we will appeal to the notion of the axes of a pyramid—the system of coordinates that best aligns with the vertices of the pyramid, such that the first coordinate aligns with the furthest stretch of the pyramid, $\cdots$, the last coordinate aligns with the shortest stretch of the pyramid. The coordinates (x_1', x_2') found above by rotation are an informal, visual example. Formally, it is necessary to define "best alignment." For convenience, we will define it in terms of least-squared distances. Then, we may use the principal component technique from multivariate statistical analysis.[5]

Definition: Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$ be a pyramid with mean vector

$$\bar{\mathbf{x}} = \frac{1}{n+1}(\mathbf{x}_1 + \cdots + \mathbf{x}_{n+1}).$$

Define the $n \times (n+1)$ matrices

$$\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}), \qquad \bar{\mathbf{X}} = (\bar{\mathbf{x}}, \cdots, \bar{\mathbf{x}}).$$

Define the $n \times n$ covariance matrix

$$\mathbf{V} = \frac{1}{n}(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})'.$$

Let $\mathbf{v}_1, \cdots, \mathbf{v}_n$ be the eigenvectors of V with eigenvalues $\lambda_1 \geqq \cdots \geqq \lambda_n$. (That is, $\mathbf{V}\mathbf{v}_i = \lambda_i \mathbf{v}_i$.) Then $\mathbf{v}_1$ is the first axis of the pyramid, $\mathbf{v}_2$ is the second axis of the pyramid, $\cdots$, $\mathbf{v}_n$ is the last axis.

The eigenvalue λ_i tells us how far the pyramid stretches along $\mathbf{v}_i$. Thus, a four-dimensional pyramid with eigenvalues $1000 \geqq 10 \geqq 10 \geqq 10$ stretches out mainly along $\mathbf{v}_1$, while a four-dimensional pyramid with eigenvalues $1000 \geqq 500 \geqq 10 \geqq 10$ stretches mainly along $\mathbf{v}_1$ and $\mathbf{v}_2$, with the stretch along $\mathbf{v}_1$ being significantly larger than the stretch along $\mathbf{v}_2$. A pyramid with identical eigenvalues does not stretch in any direction and is "fat." The eigenvalues cannot be 0 for pyramids.

The use of the axes $\mathbf{v}_1, \cdots, \mathbf{v}_n$ in evaluating a pyramid is as follows: if $\mathbf{a}$ is its direction of steepest ascent, then the ideal pyramid $\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\}$ satisfies

$$\begin{aligned} \mathbf{a}'\mathbf{v}_1 &= 0 \\ &\vdots \\ \mathbf{a}'\mathbf{v}_{n-1} &= 0. \end{aligned}$$

Thus, its first $(n-1)$ axes are perpendicular to the direction of steepest ascent, leaving the last axis, the direction of shortest stretch of the pyramid, to coincide with its direction of steepest ascent. When comparing less-than-ideal pyramids, we choose the one where $|\mathbf{a}'\mathbf{v}_i|$ is close

to 0 for as many of the first $\mathbf{v}_1, \cdots, \mathbf{v}_m$ as possible. Note that $\mathbf{a}'\mathbf{v}_i$ is the cosine of the angle between $\mathbf{a}$ and $\mathbf{v}_i$.

Example: Using the formulas given above, Table II has been completed. Note that $\mathbf{v}_1$ and $\mathbf{v}_2$ are mutually perpendicular, as they must be. From the relatively high ratios of λ_1/λ_2, it follows that all four triangles have fairly prominent major (first) axes, i.e., they stretch along some distinct direction. *ABD* and *ABE* form the narrower pair of triangles. By comparing $|\mathbf{a}'\mathbf{v}_1|$, it follows that *ADE* and *BDE* tend to stretch along their respective directions of steepest ascent, while *ABD* and *ABE* are more perpendicular to their respective steepest ascents. Therefore, we chose the triangulation $\{ABD, ABE\}$ over the triangulation $\{ADE, BDE\}$.

VII. ASSUMPTIONS

Now we spell out some assumptions about the underlying function that are relevant to the validity of our linear interpolation.

The foremost assumption is tbat the function is reasonably smooth with respect to the mesh of the interpolants. Without such basic optimism, which is buttressed by the prerogative of adding further interpolants to refine the mesh, systematic investigation could not proceed. As described in Ref. 6: "The experimenter is like a person attempting to map the depth of the sea by making soundings at a limited number of places. $\cdots$ mapping a surface resembling a nest of stalagmites or the back of a porcupine would be impossible $\cdots$ since characteristics of the surface at one point would not be related to characteristics elsewhere." Under the smoothness assumption, the axis analysis presented in Section VI is valid, since it simply recommends, in quantitative terms, the prudence of spacing out the interpolants only where the function appears to bebave uneventfully.

The face criterion of Section V applies when the function is monotone, in addition to being smooth. (Monotonicity as defined here is more stringent than what is usually required.)

Definition: A multivariate function f: $\mathbf{R}^n \to \mathbf{R}$ is monotone if one of the following holds for any $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$:

(i) For all $\alpha, \beta \in (0, 1)$, $\alpha > \beta \to f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \geqq f(\beta\mathbf{x} + (1 - \beta)\mathbf{y})$, or

Table II—Axes analysis for typical problem

Triangle	Steepest Ascent α	Eigenvectors v_1, v_2	Eigenvalues λ_1, λ_2	$\alpha' v_1$
ADE	(0.29, 0.96)′	(−0.78, −0.63)′, (0.63, −0.78)′	182.0, 59.3	−0.83
BDE	(0.99, 0.05)′	(−0.65, −0.76)′, (0.76, −0.65)′	183.7, 96.0	−0.68
ABD	(0.81, 0.58)′	(0.68, −0.73)′, (0.73, 0.68)′	233.4, 46.3	0.13
ABE	(0.98, 0.19)′	(−0.60, 0.80)′, (−0.80, −0.60)′	234.5, 75.2	−0.44

(*ii*) For all $\alpha, \beta \in (0, 1)$, $\alpha > \beta \rightarrow f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leqq f(\beta\mathbf{x} + (1 - \beta)\mathbf{y})$.

That is, f either increases or decreases consistently along whichever direction one travels in the $\mathbf{x}$-space.

Often, specific knowledge of the function includes its monotonicity. With our h, for example, it is unlikely that any direction should be singled out along which h "rollercoasts."

A further useful assumption is convexity (concavity).

Definition: A function $f: \mathbf{R}^n \rightarrow \mathbf{R}$ is convex on a convex region $A \subset \mathbf{R}^n$ if for any $\mathbf{x}, \mathbf{y} \in A$ and $\alpha \in (0, 1)$,

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leqq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y}).$$

The function f is concave if

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \geqq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y}).$$

If f is convex on the region A, the interpolated function $\hat{f}$ satisfies

$$f(\mathbf{x}) \leqq \hat{f}(\mathbf{x}) \ (\mathbf{x} \in A).$$

(Similarly, if f is concave, we have $f \geqq \hat{f}$.) Hence, knowledge of the convexity (concavity) of f permits the conclusion that the interpolated approximations are the upper (lower) bounds to the true values of f. Since h appears to be convex on the hexagon $ABCDEF$, the response time estimates obtained from linear interpolation are likely to be worst-case estimates. This agrees with actual data:

$$h(55, 10) = 1.74 \leqq 1.85 = \hat{h}(55, 10).$$

Proposition 4: Let $\|\mathbf{a}\|$ be the rate of steepest ascent in pyramid $P = \{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$. Let $r_1, \cdots, r_i$ be the rates of steepest ascent in all pyramids that share some face F with P such that $\mathbf{a}$ recedes from F. Similarly, let $s_1, \cdots, s_j$ be the rates of steepest ascent in all pyramids sharing some face with P that is approached by $\mathbf{a}$. Then, for monotone f:

(*i*) It is consistent with the convexity of f on P that

$$r_1, \cdots, r_i \leqq \|\mathbf{a}\| \leqq s_1, \cdots, s_j.$$

(*ii*) It is consistent with the concavity of f on P that

$$r_1, \cdots, r_i \geqq \|\mathbf{a}\| \geqq s_1, \cdots, s_j.$$

Example: It follows from Fig. 9 that h is likely to be convex on ABE, since $0.025 < 0.059 < 0.323$.

VIII. MORE ON THE SECOND APPROACH: A GLOBAL METRIC

Selecting the "best" system of pyramids was straightforward in our example with ABE and ADE, since the number of triangles was few

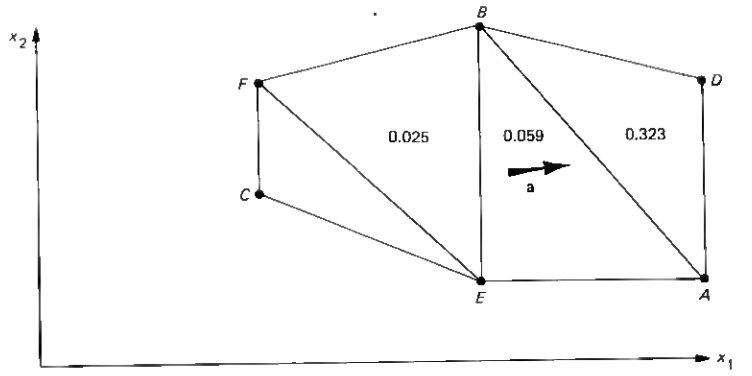

Fig. 9—Convexity of h.

and there was no trade-off between the two systems. When comparing systems of pyramids involving possibly dozens of pyramids each, and when each system contains both "good" and "bad" pyramids, individual assessments of pyramids must be combined into an overall measure to facilitate the comparison. A combined, global measure of the optimality of a system of pyramids is also necessary for the application of our procedure on a computer. We now derive such a metric under the principles discussed in Section VI.

Definition: Let $P = \{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\} \subset \mathbf{R}^n$ be a pyramid with associated steepest ascent vector $\boldsymbol{\alpha}$. Let $\lambda_1 \geqq \cdots \geqq \lambda_n > 0$ be the eigenvalues derived from P as in Section VI, and let $\mathbf{v}_1, \cdots, \mathbf{v}_n$ be the corresponding eigenvectors. Define

$$m(P) = \frac{\sum_{i=1}^{n} \lambda_i |\boldsymbol{\alpha}' \mathbf{v}_i|}{\sum_{i=1}^{n} \lambda_i}.$$

In the above expression, $m(P)$ is intended to measure the departure of P from optimality. From the weighted-average form of its definition, it follows that a large value of $m(P)$ is caused by large values of $|\boldsymbol{\alpha}' \mathbf{v}_i|$ for the larger values of λ_i. Equivalently, it implies alignment of the major axes of P along $\boldsymbol{\alpha}$. An optimal P, of course, will do the opposite: As seen in Section VI, it will align its least-significant axes along $\boldsymbol{\alpha}$.

Definition: Let $\{P_1, \cdots, P_m\}$ be a system of pyramids. Let $\mathrm{Vol}(P_j)$ denote the volume of P_j. [Recall that $\mathrm{Vol}(\{\mathbf{x}_1, \cdots, \mathbf{x}_{n+1}\}) = \det(\{\mathbf{x}_1^*, \cdots, \mathbf{x}_{n+1}^*\})/n!$; see (Ref. 7, p. 331).] Define

$$M(\{P_1, \cdots, P_m\}) = \sum_{j=1}^{m} \mathrm{Vol}(P_j) m(P_j).$$

The expression $M(\{P_1, \cdots, P_m\})$ is the proposed overall metric of the departure of the system $\{P_1, \cdots, P_m\}$ from optimality. It sums the nonoptimality of each of its constituent pyramids, giving greater weight to the more voluminous pyramids. Note that it is not necessary to normalize M by the sum

$$\sum_{j=1}^{m} \mathrm{Vol}(P_j),$$

since two comparable systems of pyramids will cover the same total volume.

Example: Applying the definitions above,

$$m(ADE) = (182.0*0.83 + 59.3*0.56)/(182.0 + 0.56) = 0.765,$$

$$\mathrm{Vol}(ADE) = \det\begin{bmatrix} 60 & 60 & 40 \\ 7 & 25 & 7 \\ 1 & 1 & 1 \end{bmatrix} = 180.$$

$$m(BDE) = 0.695, \qquad \mathrm{Vol}(BDE) = 230.$$

$$M(\{ADE, BDE\}) = 297.42;$$

$$m(ABD) = 0.269, \qquad \mathrm{Vol}(ABD) = 180.$$

$$m(ABE) = 0.492, \qquad \mathrm{Vol}(ABE) = 230.$$

$$M(\{ABD, ABE\}) = 161.85.$$

Since $M(\{ABD, ABE\}) < M(\{ADE, BDE\})$, our calculation with the metric M agrees with our earlier conclusion that $\{ABD, ABE\}$ is the preferred triangulation over $\{ADE, BDE\}$.

IX. BUILDING PYRAMIDS

One practical issue remains. In actual empirical studies, vertices become available one by one, and except when the first pyramid is formed, the pyramid building process is always applied to an existing system of pyramids. Especially in higher dimensions, we need an efficient and consistent method to incorporate new vertices into old systems of pyramids.

Example: Suppose the vertices A, B, C, D in our computer performance example have been satisfactorily triangulated, and E appears as a new vertex, as shown in Fig. 10. What new triangulations are generated so that they may be compared? If the new vertex is inside some triangle [e.g., $H = (55, 10)$ inside ABE], then an obvious reasonable answer exists (e.g., replace ABE by HBE, AHE, and ABH). In higher dimensions, the procedure is just as simple, so hereafter we will only consider new vertices not in the convex hull of the old vertices. We should specify that we start out with a minimal cover, that is, we start out

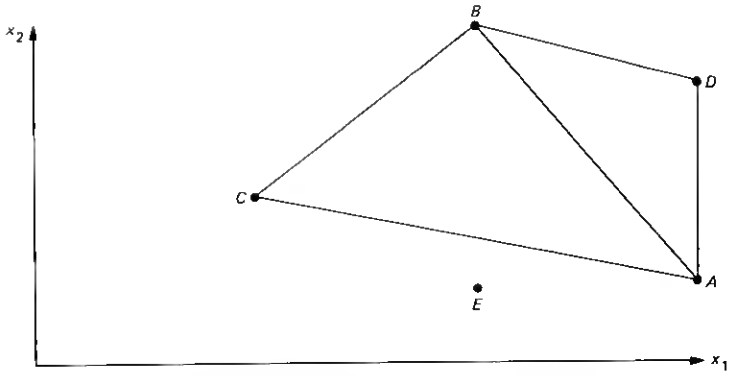

Fig. 10—Adding a new vertex to a system of pyramids.

from a system of pyramids containing the convex hull of the old points ("cover") in such a way that any point is in the interior of at most one pyramid ("minimal"). For example, if triangle *BCD* were added to the triangulation above, we would no longer have a minimal cover, since some points close to *B* would be inside two triangles. We would also like to end up with a minimal cover after incorporating the new point *E*. By going from minimal cover to minimal cover, the method we develop could be used again when a new vertex (e.g., *F*) comes around. For the sake of consistency, we do not want to add any pyramid implicitly rejected by the minimal cover we started out with, such as *BCD*. We call new covers satisfying this desideratum consistent. The appropriate building block for the new system of pyramids turns out to be faces.

Definition: Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_K\} \subset \mathbf{R}^n$ be the set of vertices from the starting minimal cover, and let $\mathbf{p}$ be a point outside its convex hull. A new face is the convex hull of $\{\mathbf{p}, \mathbf{x}_{i_1}, \cdots, \mathbf{x}_{i_{n-1}}\}$ for any sequence $1 \leqq i_1 < \cdots < i_{n-1} \leqq K$ such that:

(*i*) row rank of $(\mathbf{x}_{i_1} - \mathbf{p}, \cdots, \mathbf{x}_{i_{n-1}} - \mathbf{p}) = n - 1$, and

(*ii*) no $\mathbf{x}_j$ $(j \neq i_1, \cdots, i_{n-1})$, is in the convex hull of $\{\mathbf{p}, \mathbf{x}_{i_1}, \cdots, \mathbf{x}_{i_{n-1}}\}$.

Condition (*i*) ensures that the new face does not collapse to some $<(n-1)$ dimensional surface. It may be checked by counting the nonzero eigenvalues of the covariance matrix (see Section VI). Condition (*ii*) rules out pathologies such as face DE' in the two-dimensional case pictured in Fig. 11. (AE', of course, is legitimate.)

Proposition 5: Let F_0 be the set of old faces in the starting minimal cover, and N be the set of new faces. If S is any consistent minimal cover for the new convex hull whose set of faces is denoted by F, then

(*i*) F is a subset of $F_0 \cup N$,

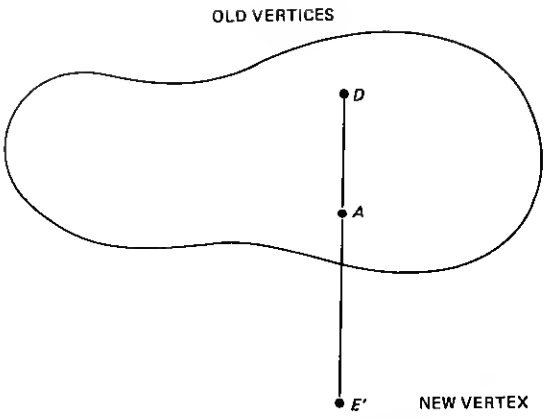

Fig. 11—The pathological triangle ADE'.

(*ii*) Any two faces from F do not intersect at interior points (though they may met at houndaries), and

(*iii*) F is a maximal set satisfying (*i*) and (*ii*), i.e., any face not in F either is not in $F_0 \cup N$ or meets some face from F at interior points.

Proof: Check that the faces from any minimal cover must satisfy (*ii*).

Thus, assuming that we can enumerate all sets of faces satisfying (*i*) through (*iii*), we may proceed to reconstruct minimal covers from these sets, and Proposition 5 tells us that we would have captured all the consistent new minimal covers we want this way. Incidentally, Proposition 5 does not rule out the possihility that we may catch more than the new minimal covers or that some sets of faces may not be reconstructible into new minimal covers. These situations are dealt with hy the converse to Proposition 5, which remains to he proven. Therefore, we should guard ourselves against the first possibility and not he surprised at the second.

The construction of all sets satisfying (*i*) to (*iii*) can he done neatly along a downward-growing hinary tree whose hranches enumerate the sets.

Example: Continuing the preceding example, it is clear that $F_0 = \{AB, AC, AD, BC, BD\}$ and $N = \{EA, EB, EC, ED\}$. Next, we list pairs of faces from $F_0 \cup N$ that meet at interior points in Tahle III. We now grow our tree. The trunk of our tree contains those faces meeting no others. Beyond that, we split the tree into two branches each time we choose hetween a face and any one of the faces it meets. We extend each branch as far as possible as long as it does not contain pairs of intersecting faces. At the end of each branch we reconstruct a new minimal cover from the faces listed on the hranch; the result is displayed in Fig. 12. Thus, the leftmost hranch $\{AD, BC, BD, EA, EC,$

Table III—Pairs of intersecting faces for typical problem

Face	Meets Face(s)
AB	*ED*
AC	*EB, ED*
AD	—
BC	—
BD	—
EA	—
EB	*AC*
EC	—
ED	*AB, AC*

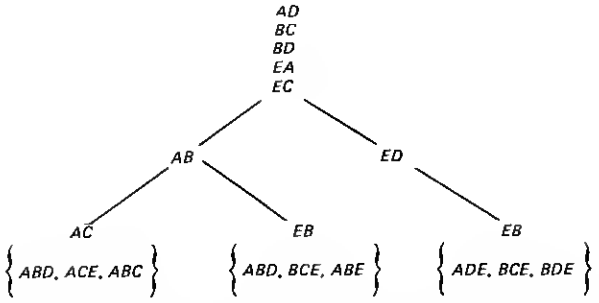

Fig. 12—Tree from new vertex E.

AB, AC} lists one of the possible sets of faces that contains as many faces as possible without including two that intersect at interior points. It is easy to see that the branch is an enumeration of all faces from the triangulation {*ABD, ACE, ABC*}. The other two branches are obtained similarly. Using the evaluation techniques from Sections V, VI, and VIII, the triangulation reconstructed from the faces listed in the middle branch was chosen.

The only step in this procedure, which is nontrivial in higher dimensions, is the tabulation of pairs of faces that intersect at interior points. Especially where there are many pairs of faces, making a pair-by-pair determination can be cumbersome.

Notation: Let $\mathbf{e}_i \in \mathbf{R}^n$ be the vector with 0 in every component except the ith, where it is 1.

Proposition 6: Let $F = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, $G = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\} \subset \mathbf{R}^n$ be faces from $F_0 \cup N$. Choose any j such that $(\mathbf{x}_1 - \mathbf{x}_n, \cdots, \mathbf{x}_{n-1} - \mathbf{x}_n, \mathbf{e}_j)$ is invertible, and let

$$\mathbf{p}'(F) = \mathbf{e}_n'(\mathbf{x}_1 - \mathbf{x}_n, \cdots, \mathbf{x}_{n-1} - \mathbf{x}_n, \mathbf{e}_j)^{-1} \in \mathbf{R}^n.$$

(If F is a face, such j exists.) Similarly, define $\mathbf{p}'(G)$. Then F and G have empty interior intersections if and only if either

(*i*) $\mathbf{p}'(F)\mathbf{x}_1 < \min\{\mathbf{p}'(F)\mathbf{y}_1, \cdots, \mathbf{p}'(F)\mathbf{y}_n\}$ or $\mathbf{p}'(F)\mathbf{x}_1 > \max\{\mathbf{p}'(F)\mathbf{y}_1, \cdots, \mathbf{p}'(F)\mathbf{y}_n\}$, or

(*ii*) $\mathbf{p}'(G)\mathbf{y}_1 < \min\{\mathbf{p}'(G)\mathbf{x}_1, \cdots, \mathbf{p}'(G)\mathbf{x}_n\}$ or $\mathbf{p}'(G)\mathbf{y}_1 > \max\{\mathbf{p}'(G)\mathbf{x}_1, \cdots, \mathbf{p}'(G)\mathbf{x}_n\}$.

Proof: The clue is to recognize that $\mathbf{p}(F)$ is simply some vector perpendicular to F and $\mathbf{p}'(F)\mathbf{y}$ is the unscaled projection of $\mathbf{y}$ onto $\mathbf{p}(F)$. Clearly, two faces have disjoint interiors if and only if the two faces have disjoint projections along some direction perpendicular to one of them.

Example: We leave the response time function h so that we may illustrate application of Proposition 6 in three dimensions. We are given six vertices in $\mathbf{R}^3$, and we wish to determine how the 20 resulting faces intersect in pairs.

$$A = \begin{pmatrix} 20 \\ 1 \\ -1 \end{pmatrix} \quad B = \begin{pmatrix} 10 \\ 1 \\ -3 \end{pmatrix} \quad C = \begin{pmatrix} 10 \\ 2 \\ -2 \end{pmatrix}$$

$$D = \begin{pmatrix} 10 \\ 5 \\ -9 \end{pmatrix} \quad E = \begin{pmatrix} 10 \\ 6 \\ -4 \end{pmatrix} \quad F = \begin{pmatrix} 9 \\ 6 \\ -6 \end{pmatrix}.$$

Table IV contains the results of the calculations according to Proposition 6. The last column of the table, where filled, lists the pairs of faces with disjoint interiors that have been identified. There are five such pairs. For example, the first row shows, using a projection perpendicular to *ABC*, that *ABC* and *DEF* do not meet, since $30 < \min\{80, 60, 69\}$. Note that *A*, *B*, and *C* have the same projection, 30, since the projection is perpendicular to *ABC*. Note also that *BCD* and *BCE* yield the same rows on Table IV. This is because *B*, *C*, *D*, and *E* lie on the same plane. By Proposition 6, the five other pairs of faces must meet at interior points.

Because only six points were involved, each row in the table could only be used to separate two three-pointed faces. If more points were involved, more than one pair of faces could be found disjoint at once. For example, if we had two additional points *G* and *H*, and the first row read as in Table V, then the following pairs of faces may immediately be identified as disjoint:

ABC, DEF	*ABH, DEF*	*AHC, DEF*
ABC, EFG	*ABH, EFG*	*AHC, EFG*
ABC, DFG	*ABH, DFG*	*AHC, DFG*
ABC, DEG	*ABH, DEG*	*AHC, DEG, ⋯*

This method clearly beats having to test each pair separately.

Table IV—Determining how 20 faces meet in pairs

Face F	$p'(F)A$	$p'(F)B$	$p'(F)C$	$p'(F)D$	$p'(F)E$	$p'(F)F$	
ABC	30.0	30.0	30.0	80.0	60.0	69.0	*ABC, DEF*
ABD	17.5	17.5	5.0	17.5	−15.0	−6.0	*ABD, CEF*
ABE	24.0	24.0	18.0	50.0	24.0	33.0	
ABF	22.2	22.2	14.4	41.0	13.2	22.2	
ACD	30.0	5.0	30.0	30.0	85.0	6.9	
ACE	30.0	60.0	30.0	140.0	30.0	69.0	*ACE, BDF*
ACF	13.3	0.0	6.7	0.0	0.0	6.0	*ACF, BDE*
ADE	22.1	12.9	14.3	22.1	22.1	21.9	*ADE, BCF*
ADF	22.2	12.8	14.4	22.2	22.6	22.2	
AEF	22.2	13.2	14.4	23.0	22.2	22.2	
BCD	20.0	10.0	10.0	10.0	10.0	9.0	
BCE	20.0	10.0	10.0	10.0	10.0	9.0	
BCF	20.3	10.5	10.5	11.8	11.3	10.5	*ADE, BCF*
BDE	20.0	10.0	10.0	10.0	10.0	9.0	
BDF	20.1	9.7	10.2	9.7	11.1	9.7	*ACE, BDF*
BEF	20.4	11.4	10.8	14.0	11.4	11.4	
CDE	20.0	10.0	10.0	10.0	10.0	9.0	
CDF	20.3	9.9	10.5	10.5	11.9	10.5	
CEF	20.3	11.3	10.5	13.3	10.5	10.5	*ABD, CEF*
DEF	23.0	14.0	16.0	27.0	27.0	27.0	*ABC, DEF*

Table V—Separating many pairs of faces simultaneously

Face F	$p'(F)A$	$p'(F)B$	$p'(F)C$	$p'(F)D$	$p'(F)E$	$p'(F)F$	$p'(F)G$	$p'(F)H$
ABC	30	30	30	80	60	69	70	25
.								
.								
.								

X. RECAPITULATION

The cycle is now complete. We start out with some minimal cover—possibly consisting of a single pyramid—on the **x**-space of some function $y = f(\mathbf{x})$. The behavior of the function on the minimal cover may be explored by linear interpolation as discussed in Section II. If further information is subsequently found wanting, a new interpolant is chosen, and a new data point is empirically acquired. Using the technique from the preceding section, we are able to list *all* consistent ways to extend the starting minimal cover into a new minimal cover that incorporates the new interpolant. These new minimal covers should be evaluated both in terms of their faces and in terms of the axes of their pyramids. For either evaluation, the basic ingredient is the notion of steepest ascent. Through this notion the values of the function at the interpolants are taken into account. This distinguishes our approach from other schemes where only the **x**-space gets examined. Eventually, a best new minimal cover is chosen, and we are ready, once more, to interpolate and explore the function.

REFERENCES

1. R. A. Becker, and J. M. Chambers, "Design and Implementation of the *S* System for Interactive Data Analysis," Proc. Computer Software and Applications Chi-

cago, Illinois (November 13–16, 1978), pp. 626–9.
2. C. L. Lawson, "Generation of a Triangular Grid With Application to Contour Plotting," Tech. Memo. 299, Sect. 914, Jet Propulsion Lab., Calif. Inst. Tech. (February 1972).
3. H. A. Akima, "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points," ACM Trans. Math. Software, *4*, No. 2 (June 1978), pp. 148–59.
4. W. E. Milne et al., "Mathematics for Digital Computers," Vol. 1: Multivariate Interpolation, Wright Air Development Center Tech. Rep. 57–556, Aeronautical Research Laboratory (February 1958).
5. T. W. Anderson, *An Introduction of Multivariate Statistical Analysis*, New York: John Wiley, 1974.
6. G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*, New York: John Wiley, 1978.
7. G. Birkhoff and S. MacLane, *A Survey of Modern Algebra*, 4th ed., New York: McMillan Publishing Co., 1977.